# Three-site quantum lattice with thermal bath

Paolo M. Pumilia

February 1, 2008

paolo.pumilia@acm.org

august 19th, 2002



**Abstract**

An algorithm to simulate the dynamics of a quantum state over a three-site lattice interacting with classical harmonic oscillators has been devised.

The oscillators are linearly coupled to the quantum state and are acted upon by a fluctuation-dissipation process to take the equilibrium thermal environment into account, thus allowing to investigate how stochastic forces may affect the quantum dynamics.

The implementation of the algorithm has been written in Ada95.

## 1 The physical model

The Hosltein hamiltonian for a lattice connected to independent site oscillators can be distinguished into three terms

$$h = h_a + h_b + h_{ab} \tag{1}$$

corresponding to the *primary* system $a$, representing those states allowed to the quantum particle, the *secondary* system $b$, representing the vibrating lattice masses ($m$) and the interaction energy $ab$ [?].

In a three-site lattice, assuming the nearest neighbours approximation hold, the quantum states are described by the hamiltonian

$$h_a = V\left(|1\rangle\langle 2| + |2\rangle\langle 1| + |2\rangle\langle 3| + |3\rangle\langle 2|\right) \tag{2}$$

where $|n\rangle$ are the eigenstates of the isolated sites and $V$ is the overlap integral between neighbour sites.

Vibrations are described by harmonic oscillators of equal mass $m$ and frequency $\omega$, with respective position $r_n$ and conjugated variable $p_n$, (where $n$ indicates the location on the lattice):

$$h_b = \frac{1}{2m}\left(p_1^2 + m^2\omega^2 r_1^2\right) + \frac{1}{2m}\left(p_2^2 + m^2\omega^2 r_2^2\right) + \frac{1}{2m}\left(p_3^2 + m^2\omega^2 r_3^2\right) \tag{3}$$

Interaction terms are assumed to be linear with respect to $r_n$

$$h_{ab} = \epsilon_1 |1\rangle\langle 1| + \epsilon_2 |2\rangle\langle 2| + \epsilon_3 |3\rangle\langle 3| \tag{4}$$

$\epsilon_n = \chi r_n$ , with $\chi$ real positive, being the coupling energy at site $n$. Such physical model has been studued by Hennig [?] and Kenkre & Andersen [?], but without thermal environment.

## 1.1 Lattice with free boundary conditions

When loose ends boundary conditions apply to the primary system, we are lead to the following equation for the density matrix:

$$\begin{aligned}
\dot{\rho}_{11} &= i\omega_o\left(\rho_{21}-\rho_{12}\right)\\
\dot{\rho}_{12} &= -i\omega_{12}\rho_{12}+i\omega_o\left(\rho_{22}-\rho_{11}-\rho_{13}\right)\\
\dot{\rho}_{13} &= -i\omega_{13}\rho_{13}+i\omega_o\left(\rho_{23}-\rho_{12}\right)\\
\dot{\rho}_{21} &= i\omega_{12}\rho_{21}+i\omega_o\left(\rho_{11}+\rho_{31}-\rho_{22}\right)\\
\dot{\rho}_{22} &= i\omega_o\left(\rho_{12}+\rho_{32}-\rho_{21}-\rho_{23}\right)\\
\dot{\rho}_{23} &= -i\omega_{23}\rho_{23}+i\omega_o\left(\rho_{13}+\rho_{33}-\rho_{22}\right)\\
\dot{\rho}_{31} &= i\omega_{13}\rho_{31}+i\omega_o\left(\rho_{21}-\rho_{32}\right)\\
\dot{\rho}_{32} &= i\omega_{23}\rho_{32}+i\omega_o\left(\rho_{22}-\rho_{31}-\rho_{33}\right)\\
\dot{\rho}_{33} &= i\omega_o\left(\rho_{23}-\rho_{32}\right)
\end{aligned} \tag{5}$$

where $\omega_o = V/h$ is the frequency for the oscillation of the quasi-particle between two neighbour sites and $\omega_{kn} = -\omega_{nk} = \left(\epsilon_n - \epsilon_k\right)/\hbar = \chi\left(r_n - r_k\right)/\hbar$.

Using operators

$$\begin{aligned}
u_1 &= \rho_{11}-\rho_{22}\\
u_2 &= \rho_{22}-\rho_{33}\\
v_1 &= i\left(\rho_{12}-\rho_{21}\right)\\
v_2 &= i\left(\rho_{32}-\rho_{23}\right)\\
v_3 &= i\left(\rho_{31}-\rho_{13}\right)\\
w_1 &= \rho_{12}+\rho_{21}\\
w_2 &= \rho_{23}+\rho_{32}\\
w_3 &= \rho_{13}+\rho_{31}
\end{aligned} \tag{6}$$

a new set of equations, that will be used to perform numerical simulations, can be obtained for the primary system.

As a fluctuation-dissipation process has been finally attached to each oscillator, the complete dynamical model can be written:

$$\begin{aligned}
\dot{u}_1 &= -\omega_o\left(2v_1+v_2\right)\\
\dot{u}_2 &= \omega_o\left(v_1+2v_2\right)\\
\dot{v}_1 &= \omega_{12}w_1+\omega_o\left(2u_1+w_3\right)\\
\dot{v}_2 &= -\omega_{23}w_2-\omega_o\left(2u_2-w_3\right)\\
\dot{v}_3 &= -\omega_{13}w_3-\omega_o\left(w_1-w_2\right)\\
\dot{w}_1 &= -\omega_{12}v_1+\omega_o v_3\\
\dot{w}_2 &= \omega_{23}v_2-\omega_o v_3\\
\dot{w}_3 &= \omega_{13}v_3-\omega_o\left(v_1+v_2\right)\\
\dot{r}_1 &= p_1/m\\
\dot{r}_2 &= p_2/m\\
\dot{r}_3 &= p_3/m\\
\dot{p}_1 &= -m\omega^2 r_1-\frac{\chi}{3}\left(c+u_2+2u_1\right)-\gamma_1 p_1+f_1\left(t\right)\\
\dot{p}_2 &= -m\omega^2 r_2-\frac{\chi}{3}\left(c+u_2-u_1\right)-\gamma_2 p_2+f_2\left(t\right)\\
\dot{p}_3 &= -m\omega^2 r_3-\frac{\chi}{3}\left(c-u_1-2u_2\right)-\gamma_3 p_3+f_3\left(t\right)
\end{aligned} \tag{7}$$

where $\gamma_n$ are the damping coefficients and $f_n\left(t\right)$ models gaussian noise, with $\delta$-shaped time correlation, satisfying the following

relations:

$$\begin{aligned} \langle f_n(t)\rangle &= 0 \quad (8)\\ \langle f_n(t) f_n(t')\rangle &= 2\gamma\theta\delta_{(t-t')} \quad (9)\end{aligned}$$

having defined thermal energy

$$\theta = k_B T \quad (10)$$

($k_B$being the Boltzmann constant); hence

$$\left\langle f_n^2(t)\right\rangle^{1/2} = (2\gamma\theta)^{1/2} \quad (11)$$

It can be readily verified that

$$c = \rho_{11} + \rho_{22} + \rho_{33} \quad (12)$$

is a constant of motion. A further constant, that will be used to monitor numerical simulations, is given by

$$K = \frac{4}{3}\left(u_1^2 + u_2^2 + u_1 u_2\right) + v_1^2 + v_2^2 + v_3^2 + w_1^2 + w_2^2 + w_3^2 \quad (13)$$

The dynamical problem is described by a system of stochastic real valued, first order, differential equations (7) that must be numerically solved.

Notice that, restricting the number of sites to $n = 1, 2$ we are left with the spin-boson equation with noise:

$$\begin{aligned} \dot{u}_1 &= -2\omega_o v_1\\ \dot{v}_1 &= \omega_{12} w_1 + 2\omega_o u_1\\ \dot{w}_1 &= -\omega_{12} v_1\\ \dot{r}_1 &= p_1/m\\ \dot{r}_2 &= p_2/m\\ \dot{p}_1 &= -m\omega^2 r_1 - \frac{\chi}{3}\left(c + u_2 + 2u_1\right) + f_1(t)\\ \dot{p}_2 &= -m\omega^2 r_2 - \frac{\chi}{3}\left(c + u_2 - u_1\right) + f_2(t)\end{aligned} \quad (14)$$

that has been studied by many researchers; in particular, the author's work has been inspired the papers[**?**, **?**], referenced to in the bibliography.

## 2 Numerical integration method

Stochastic differential equations of first order in time for a vector variabile $\mathbf{x}(t)$ take the from,

$$\dot{\mathbf{x}} = \mathbf{F} + \mathbf{g}\mathbf{f} \quad (15)$$

where $\mathbf{F} = (F_i)_{1,n} = (F_i(\mathbf{x}, t))_{1,n}$ is the vector of deterministic fields, $\mathbf{f} = (f_i)_{1,n} = (f_i(t))_{1,n}$ the gaussian stochastic force, satisfying the conditions:

$$\begin{aligned} \langle f(t)\rangle &= 0\\ \langle f(t) f(t')\rangle &= \delta(t - t') \quad (16)\end{aligned}$$

while th parameters set $\mathbf{g} = (g_i)_{1,n} = (g_i(\mathbf{x}, t))_{1,n}$ represents the interaction of the system with the thrmal bath.

Equations given in (15) are *entirely coupled,* since, in general, every components $F_i$ and $g_i$ are functions of the whole variable $\mathbf{x}$,

The formal solution of (15), assuming the *implicit* time dependence $F = \mathbf{F}(\mathbf{x}(t))$ e $g = \mathbf{g}(\mathbf{x}(t))$, is given by [**?**]

$$x_i(t) - x_i(0) = \int_0^t dt' F_i + \int_0^t dt' g_i f_i \qquad i = 1, \ldots n \quad (17)$$

Given a time interval $[0,h]$ that can be considered infinitely small for $F_i$ e $g_i$ functions, their value at the arbitrary instant $t' \in [0,h]$, can be approximated by the first $\kappa$ terms of a Taylor expansion, centered at $t=0$:

$$F_i(t) = F_i^o + \delta^\kappa F_i \tag{18}$$
$$g_i(t) = g_i^o + \delta^\kappa g_i \tag{19}$$

The index $\kappa$ in (18,19) indicates the truncation order of the series:

$$\delta F_i = \sum_j [F_i]_j \cdot \delta x_j(t') + \frac{1}{2}\sum_{jk} [F_i]_{jk} \cdot \delta x_j(t')\delta x_k(t') + \frac{1}{3!}\sum_{jk} [F_i]_{jkl} \cdot \delta x_j(t')\delta x_k(t')\delta x_l + \ldots \tag{20}$$

$$\delta g_i = \sum_j [g_i]_j \cdot \delta x_j(t') + \frac{1}{2}\sum_{jk} [g_i]_{jk} \cdot \delta x_j(t')\delta x_k(t') + \frac{1}{3!}\sum_{jk} [g_i]_{jkl} \cdot \delta x_j(t')\delta x_k(t')\delta x_l + \ldots \tag{21}$$

in which $F_i$ and $g_i$, time derivatives evaluated at $t=0$, have been represented by the following notation:

$$[F_i]_j \equiv \frac{\partial F_i}{\partial x_j}(0) \quad [F_i]_{jk} \equiv \frac{\partial^2 F_i}{\partial x_j \partial x_k}(0) \quad [F_i]_{jkl} \equiv \frac{\partial^3 F_i}{\partial x_j \partial x_k \partial x_l}(0) \tag{22}$$

$$[g_i]_j \equiv \frac{\partial g_i}{\partial x_j}(0) \quad [g_i]_{jk} \equiv \frac{\partial^2 g_i}{\partial x_j \partial x_k}(0) \quad [g_i]_{jkl} \equiv \frac{\partial^3 g_i}{\partial x_j \partial x_k \partial x_l}(0) \tag{23}$$

Truncation at $\kappa = 0$ the general solution obtained after substitution of (18,19) into (17):

$$x_i(h) - x_i(0) = \int_0^h dt' \left(F_i^o + \delta^\kappa F_i\right) + \int_0^h dt' \left(g_i^o + \delta^\kappa g_i\right) f_i \tag{24}$$

yields

$$x_i(h) - x_i(0) = F_i^o h + g_i^o \int_0^h dt' f_i \tag{25}$$

where $F_i = [F_i] = F_i^o$ and $g_i = [g_i] = g_i^o$ have been defined.

Since the gaussian integral

$$Z_1(h) = \int_0^h dt' f(t') \tag{26}$$

is of $h^{1/2}$ order, the second term in (25) is the lowest order approximation of the trajectory (24):

$$\delta x_i^{(1/2)} = g_i^o Z_1 \tag{27}$$

having defined $\delta x_i = x_i(h) - x_i(0)$.

Substituting (27) into (20,21), hence into (17), and retaining only contribution of the order $h$ in (24), we obtain the first order correction

$$\begin{aligned}
\delta x_i^{(1)} &= F_i^o h + \int_0^h dt' \sum_j [g_i]_j \cdot \delta x_j^{(1/2)} f_i(t') \\
&= F_i^o h + \sum_j [g_i]_j \cdot g_j^o \int_0^h dt' Z_1(t') f_i(t') \\
&= F_i^o h + \frac{1}{2}\sum_j [g_i]_j \cdot g_j^o Z_1^2
\end{aligned} \tag{28}$$

The corresponding displacement on the trajectory is thus given by

$$\delta x_i = \delta x_i^{1/2} + \delta x_i^1 \tag{29}$$

Again substituting the first order correction (28) into the series (20,21), hence into (17), then retaining only terms of the order $h^{3/2}$ in (24), we get:

$$\begin{aligned}
\delta x_i^{(3/2)} &= \sum_j [F_i]_j \cdot g_j^o Z_1 h + \int_0^h dt' \sum_j [g_i]_j \cdot \delta x_j^{(1)} f_i(t') \\
&= \sum_j [F_i]_j \cdot g_j^o Z_1 h + \int_0^h dt' \sum_j [g_i]_j \cdot \left( F_i^o h + \frac{1}{2} \sum_k [g_i]_k \cdot g_k^o Z_1^2(t') f_i(t') \right) \\
&= \sum_j [F_i]_j \cdot g_j^o Z_1 h + \sum_j [g_i]_j \cdot \left( F_i^o h + \frac{1}{2} \sum_k [g_i]_k \cdot g_{k'}^o \int_0^h dt Z_1^2(t') f_i(t') \right) \\
&= \sum_j [F_i]_j \cdot g_j^o Z_1 h + \sum_j [g_i]_j \cdot \left( F_i^o h + \frac{1}{3!} \sum_k [g_i]_k \cdot g_{k'}^o Z_1^3 \right)
\end{aligned} \tag{30}$$

In the simplified case in which $g$ does not depend on $\mathbf{x}$, the numerical algorithm to reach the order $h^2$ in approximating the exact solution of (15) can then be written

$$x_i(h) - x_i(0) = g_i^o Z_1 + F_i^o h + \sum_j [F_i]_j \cdot g_j^o Z_2 + \frac{1}{2} \sum_j [F_i]_j \cdot [F_j] \, h^2 \qquad i = 1, \ldots n \tag{31}$$

or, in vectorial notation:

$$\mathbf{x}(h) - \mathbf{x}(0) = \mathbf{g}^o \mathbf{Z}_1 + \mathbf{F}^o \mathbf{h} + \nabla \mathbf{F} \mathbf{g} \mathbf{Z}_2 + \frac{1}{2} \nabla \mathbf{F} \cdot \mathbf{F} h^2 \tag{32}$$

## 2.1 The algorithm for the three-site lattice

Equations (7) belong to the class (15); therefore the approximation method that has been recalled in the previous chapter can be implemented.

To better expose our procedure, it is convenient switching to the vectorial notation; the system variable $\mathbf{x}$, taking the form:

$$\mathbf{x} = \begin{bmatrix} u_1 \\ u_2 \\ v_1 \\ v_2 \\ v_3 \\ w_1 \\ w_2 \\ w_3 \\ r_1 \\ r_2 \\ r_3 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} \tag{33}$$

evolves in time according to the equation

$$\dot{\mathbf{x}} = \mathbf{F} + \mathbf{G} \tag{34}$$

where $\mathbf{F}$ is the vector of the deterministic forces

$$\mathbf{F} = \begin{bmatrix} -\omega_o\left(2v_1+v_2\right) \\ \omega_o\left(v_1+2v_2\right) \\ \omega_{12}w_1+\omega_o\left(2u_1+w_3\right) \\ -\omega_{23}w_2-\omega_o\left(2u_2-w_3\right) \\ -\omega_{13}w_3-\omega_o\left(w_1-w_2\right) \\ -\omega_{12}v_1+\omega_o v_3 \\ \omega_{23}v_2-\omega_o v_3 \\ \omega_{13}v_3-\omega_o\left(v_1+v_2\right) \\ p_1/m \\ p_2/m \\ p_3/m \\ -m\omega^2 r_1-\frac{\chi}{3}\left(c+u_2+2u_1\right)-\gamma_1 p_1 \\ -m\omega^2 r_2-\frac{\chi}{3}\left(c+u_2-u_1\right)-\gamma_2 p_2 \\ -m\omega^2 r_3-\frac{\chi}{3}\left(c-u_1-2u_2\right)-\gamma_3 p_3 \end{bmatrix} \tag{35}$$

while $\mathbf{G}$ contains the average values of the stochastic forces:

$$\mathbf{G} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \left(2\gamma_1\theta_1\right)^{1/2} \\ \left(2\gamma_2\theta_2\right)^{1/2} \\ \left(2\gamma_3\theta_3\right)^{1/2} \end{bmatrix} \tag{36}$$

The approximated solution up to the second order in time can be computed according to the formula (32):

$$\mathbf{x}(h) \simeq \mathbf{x}(0)+\mathbf{g}\cdot\mathbf{Z}_1+\mathbf{F}\cdot\mathbf{h}+\nabla\mathbf{F}\cdot\mathbf{g}\cdot\mathbf{Z}_2+\frac{1}{2}\nabla\mathbf{F}\cdot\mathbf{F}\cdot\mathbf{h}^2 \tag{37}$$

where $Z_1$ and $Z_2$ are gaussian integrals and the force gradient is given by:

$$\nabla\mathbf{F} = \begin{bmatrix} 0 & 0 & -2\omega_o & -\omega_o & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \omega_o & 2\omega_o & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2\omega_o & 0 & 0 & 0 & 0 & \omega_{12} & 0 & \omega_o & -\chi_1 w_1 & \chi_2 w_1 & 0 & 0 & 0 & 0 \\ 0 & -2\omega_o & 0 & 0 & 0 & 0 & -\omega_{23} & \omega_o & 0 & \chi_2 w_2 & -\chi_3 w_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\omega_o & \omega_o & -\omega_{13} & \chi_1 w_3 & 0 & -\chi_3 w_3 & 0 & 0 & 0 \\ 0 & 0 & -\omega_{12} & 0 & \omega_o & 0 & 0 & 0 & \chi_1 v_1 & -\chi_2 v_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \omega_{23} & -\omega_o & 0 & 0 & 0 & 0 & -\chi_1 v_2 & \chi_3 v_2 & 0 & 0 & 0 \\ 0 & 0 & -\omega_o & -\omega_o & \omega_{13} & 0 & 0 & 0 & -\chi_1 v_3 & 0 & \chi_3 v_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/m_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/m_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/m_2 \\ -2\chi_1/3 & -\chi_1/3 & 0 & 0 & 0 & 0 & 0 & 0 & -m_1\omega_1^2 & 0 & 0 & -\gamma_1 & 0 & 0 \\ \chi_2/3 & -\chi_2/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -m_2\omega_2^2 & 0 & 0 & -\gamma_2 & 0 \\ \chi_3/3 & 2\chi_3/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -m_3\omega_3^2 & 0 & 0 & -\gamma_3 \end{bmatrix} \tag{38}$$

The implementation of the algorithm according to the rule (37) is displayed in the last section . Ada95 language has been the language of choiche, since it is maintained that a rather easy and dependable parallel code, that could be required to perform long lasting simulations can be worked out in that environment, .

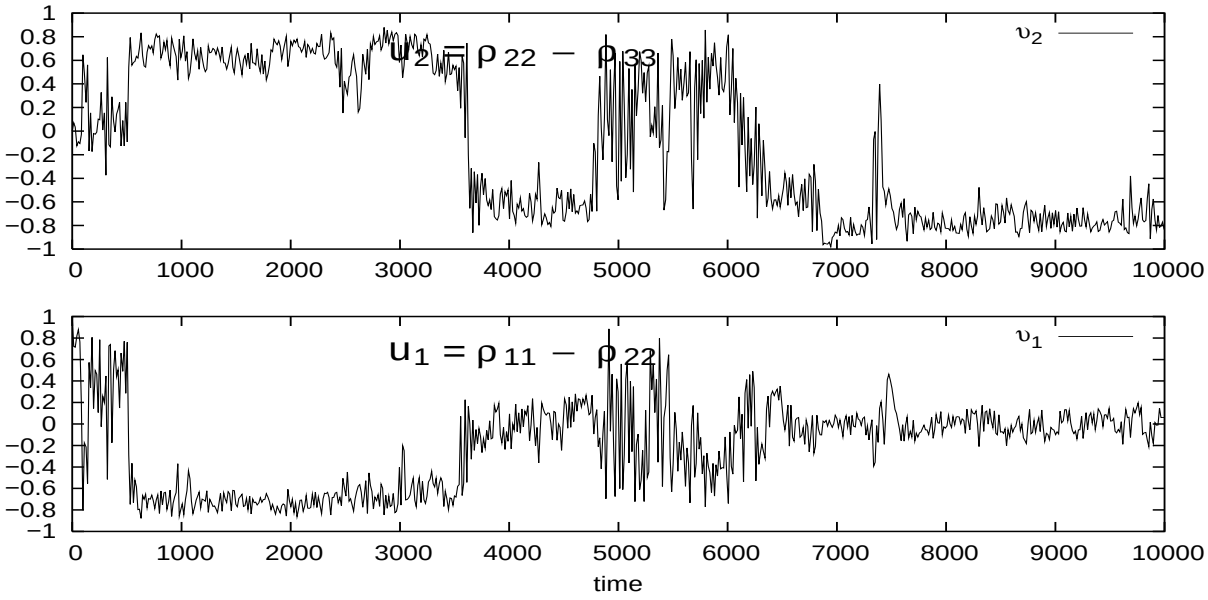
Marsaglia & Tsang [?] fortran routine for random numbers generation has been used to compute $Z_1$ and $Z_2$ integrals.

# 3 Results and Conclusions

First run of the program allowed to plot variables : $u_1 = \rho_{11} - \rho_{22}$ and $u_2 = \rho_{22} - \rho_{33}$, using the set of physical parameters:

- inter-site interaction $V = 0.09$
- free oscillators frequencies $\omega^2 = 2x10^{-3}$
- vibration-quantum state coupling $\chi = 4x10^{-2}$ ;
- $\gamma = 0.2$ that imply $\omega^2/2\gamma^2 = 0.025$ and $\chi^2/2m\omega^2 = 0.1$

Time integration step has been set to $10^{-3}$ and the reliability of the computation has been monitored through the constant (13), whose changes have kept below $10^{-7}$.




(a) occupation differences $u_1 = \rho_{11} - \rho_{22}$ (lower) and $u_2 = \rho_{22} - \rho_{33}$ (upper) evolving in time

Picture of the site occupation differences $u_1 = \rho_{11} - \rho_{22}$ (lower) and $u_2 = \rho_{22} - \rho_{33}$ (upper) evolving in time (arbitrary units) shows the localized motion of the quantum state.

The possibilty to model the dynamics of a three-site lattice coupled to stochastic oscillators allows one to study how a quantum state propagates in a real lattice, when vibrational motion can be described classically, then to investigate the duration of out of equilibrium states and the conditions under which energy transfer without dissipation may occur.

# 4 Code listing

Here included are the main tokens of code from the iterator procedure, in which the numerical integration at time step `dh` is performed:

```
procedure iterator (x : in out coordinates ; xh : out coordinates ;                                                    time_step   : in double
                    outer_cycle : in integer ; -- total number of trajectory   -- points to print out

v,am1,am2,am3,gam1,gam2,gam3,om1,om2,om3: in double;                                          chi1,chi2,chi3,t1,t2,t3,e1b,e2b,e3b  : in do
                    output_file :  in string                )
is
```

Fortran random numbers generation routine `rnor` is imported here:

```
function rnor(dum : in Fortran.Fortran_Integer) return Fortran.double_precision;
pragma Import (fortran, rnor, "rnor_");
```

Definition of numerical parameters:

```
dh  : double := time_step ; -- time integratin step
dh2 : double := time_step * time_step; --  squared time integration step
dh25 := dh2 * 0.5;
```

Definition of physical parameters:

```
v2:   double := v * v ; --
ak0 := c1*c1 + c2*c2 + c3*c3;
qom1 := om1*om1;
qom2 := om2*om2;
qom3 := om3*om3;
qgam1 := gam1 * gam1;
qgam2 := gam2 * gam2;
qgam3 := gam3 * gam3;
norm := (1.0/3.0)*( x(1)*x(1) + x(2)*x(2) + x(1)*x(2) );
norm := norm + x(3)*x(3) + x(4)*x(4) + x(5)*x(5);
norm := norm + x(6)*x(6) + x(7)*x(7) + x(8)*x(8);
norm := (3.0/4.0) * norm;
```

Here the iteration starts:

```
for i in 1..outer_cycle loop
  for j in 1..inner_cycle loop
```

Definition of some useful expressionss:

```
    time := time + time_step;
    w11 := double(rnor(dum));
    w12 := double(rnor(dum));
    w21 := double(rnor(dum));
    w22 := double(rnor(dum));
    w31 := double(rnor(dum));
    w32 := double(rnor(dum));
    z11 := a11*w11 ;
    z12 := a21*z11 + a22*w12;
    z21 := a11*w21 ;
    z22 := a21*z21 + a22*w22;
    z31 := a11*w31 ;
    z32 := a21*z31 + a22*w32;
    om12 := de12  - chi2*x(10) + chi1*x(9);
    om23 := de23  - chi3*x(11) + chi2*x(10);
    om13 := de13  - chi3*x(11) + chi1*x(9);
    qom12 := om12 * om12;
    qom13 := om13 * om13;
    qom23 := om23 * om23;
    akey21 := - chi2*x(13)/am2 + chi1*x(12)/am1 ;
    akey23 := - chi2*x(13)/am2 + chi3*x(14)/am3 ;
    akey31 := - chi3*x(14)/am3 + chi1*x(12)/am1 ;
    ajay1 := chi1*(ak0+x(1)/2.0+x(2))/3.0;
    ajay2 := chi2*(ak0+x(1)/2.0-x(2)/2.0)/3.0;
    ajay3 := chi3*(ak0-x(1)-x(2)/2.0)/3.0;
    omx3 := om23 + om13;
    omx1 := om12 + om13;
```

Variables incrementation:

```
  xh(1) := x(1) + 2.0*v*( 2.0*x(7) + x(6) ) * dh +
                  v2*(x(2)-2.0*x(1)+3.0*x(5)) * dh2 +
                  v*(2.0*om23*x(4) - om12*x(3)) * dh2;

  xh(2) := x(2) - 2.0*v*( x(7) + 2.0*x(6) ) * dh +
                  v2*(x(1)-2.0*x(2)-3.0*x(5)) * dh2 +
                  v*(2.0*om12*x(3) - om23*x(4)) * dh2;

  xh(3) := x(3) + ( om12*x(6) + v*x(8) ) * dh -
                    qom12*x(3) * dh25 + v2*(x(4)-x(3)) * dh25 +
                    v*(om12*x(2) + omx1*x(5)) * dh25 +
                    akey21*x(6) * dh25 ;

  xh(4) := x(4) - ( om23*x(7) + v*x(8) ) * dh -
                    qom23*x(4) * dh25 + v2*(x(3)-x(4)) * dh25 +
                    v*(om23*x(1) - omx3*x(5)) * dh25 +
                    akey23*x(7) * dh25 ;

  xh(5) := x(5) - ( om13*x(8) + v*x(6) + v*x(7) ) * dh -
                    qom13*x(5) * dh25 + v2*(x(1)-x(2) - 2.0*x(5)) * dh25 +
                    v*(omx1*x(3) - omx3*x(4)) * dh25 -
                    akey31*x(8) * dh25 ;

  xh(6) := x(6) - ( om12*x(3) - v*x(2) - v*x(5) ) * dh -
                    qom12*x(6) * dh25 - v2*(5.0*x(6)+3.0*x(7)) * dh25 -
                    v*omx1*x(8) * dh25 -
                    akey21*x(3) * dh25 ;

  xh(7) := x(7) + ( om23*x(4) - v*x(1) + v*x(5) ) * dh -
                    qom23*x(7) * dh25 - v2*(3.0*x(6)+5.0*x(7)) * dh25 -
                    v*omx3*x(8) * dh25 -
                    akey23*x(4) * dh25 ;

  xh(8) := x(8) + ( om13*x(5) - v*x(3) + v*x(4) ) * dh -
                    qom13*x(8) * dh25 - 2.0*v2*x(8) * dh25 -
                    v*(omx1*x(6) + omx3*x(7)) * dh25 +
                    akey31*x(5) * dh25 ;

  xh(9) := x(9) + (x(12)/am1) * dh -
                  qom1*x(9) * dh25 -
                  dh25 * (ajay1 + gam1*x(12))/am1 +
                  (D1/am1) * z12  ;

  xh(10) := x(10) + (x(13)/am2) * dh -
                     qom2*x(10) * dh25 -
                     dh25 * (ajay2 + gam2*x(13))/am2 +
                     (D2/am2) * z22  ;

  xh(11) := x(11) + (x(14)/am3) * dh -
                     qom3*x(11) * dh25 -
                     dh25 * (ajay3 + gam3*x(14))/am3 +
                     (D3/am3) * z32  ;

  xh(12) := x(12) + D1*z11 -
                    ( am1*qom1*x(9) + ajay1 + gam1*x(12) ) * dh +
                      qom1*(am1*gam1*x(9) - x(12)) * dh25 +
                     (chi1*v*x(6) + gam1*ajay1 + qgam1*x(12)) * dh25 -
                      gam1*D1*z12;

  xh(13) := x(13) + D2*z21 -
                   ( am2*qom2*x(10) + ajay2 + gam2*x(13) )*dh +
                   qom2*(am2*gam2*x(10) - x(13)) * dh25 +
                   (-chi2*v*(x(7)+x(6)) + gam2*ajay2 +  qgam2*x(13)) * dh25 -
                    gam2*D2*z22;

  xh(14) := x(14) + D3*z31 -
                   ( am3*qom3*x(11) + ajay3 + gam3*x(14) )*dh +
                    qom3*(am3*gam3*x(11) - x(14)) * dh25 +
                    (chi3*v*x(7) + gam3*ajay3 + qgam3*x(14)) * dh25 -
                   gam3*D3*z32;

      for ii in 1..config_space  loop
        x(ii) := xh(ii);
      end loop;

end loop;
```

```
end loop;
```